

# A Trustworthy Distributed Social Carpool Method

Francisco Martín-Fernández<sup>(✉)</sup>, Cándido Caballero-Gil,  
and Pino Caballero-Gil

Department of Computer Engineering,  
University of La Laguna, San Cristóbal de La Laguna, Spain  
{fmartinf, ccabgil, pcaballe}@ull.edu.es

**Abstract.** Due to the increase in the number of vehicles over the years, roads are witnesses of frequent traffic congestions, especially in urban environments. The problem is even worse because most vehicles are occupied by a single person. Therefore, a possible solution would be to use carpooling to increase the number of people in vehicles. In this way, commuters who travel from and to nearby places will share the same vehicle. Existing carpool solutions are unpopular for most people because of the distrust generated by carpooling with strangers. Thanks to the widespread of mobile technology and social networks, it is possible to create a reliable, friendly and secure carpool system. This paper proposes the design and implementation of an integrated solution that can be used to improve existing carpool solutions. The proposal is based on a novel algorithm that rates user behaviours based on interactions and friendship degrees. In addition to this, the whole carpool system has been developed as a mobile application for Android devices, and the Open Source Project is already available on the Github platform.

## 1 Introduction

The number of vehicles in the world is growing at a rapid pace due to the rising of global population. This fact causes a dramatic rise in fuel prices and pollution. Therefore, governments have been forced into drastic measures to combat pollution, especially in large urban centers [15, 22, 25]. One of the solutions proposed by the vast majority of governments is the encouraging of use of public transport. Unfortunately, this solution is not convenient or comfortable enough for many users because of the existing constraints in time or place.

A convenient solution involves sharing cars so that empty seats can be used in most trips. This modality is known with the term carpooling and has been proposed as an effective way to reduce both pollution and spending. Another related but different proposal, known as carsharing, is based on collective fleets of cars with multiple users, but such a solution does not solve as many problems as carpooling. Ridesharing is the general term used to refer to solutions for sharing the use of a car with other people in order to travel to a given destination. Apart from carsharing and carpooling (also known as real-time or instant or dynamic or

ad-hoc ridesharing), ridesharing also includes other versions known as slugging, lift sharing and covoiturage. However, ridesharing proposals that are different from carpooling are out of the scope of this work.

The use of both types of collaborative solutions has increased since the onset of the economic crisis thanks to technology 2.0 [18]. They are applicable in almost any environment, but are especially useful in places like universities, holidays, long journeys and even urban centres. This is because in these situations both vehicle owners and passengers agree on the same motivations to consider carpooling. Usually their main goal is to share fuel cost, but there may be other reasons such as solving the parking problem, wanting to talk with others or to take care of the environment, etc.

This work proposes an improvement for existing carpooling solutions that make use of recent technological advances of smartphones and social networks. The described solution allows the establishment of trust and reputation accountability between drivers and passengers, while taking care of their privacy at the same time.

This paper is organized as follows. Section 2 mentions several related works. Section 3 introduces the general design of the proposal, which is mainly based on the reputation algorithm sketched in Sect. 4. The security of the proposal is briefly analyzed in Sect. 5. Section 6 describes the developed Android application. Finally, some conclusions and open problems can be found in Sect. 7.

## 2 Related Work

The first carpooling projects emerged in the late 1980s [24], but in those days, without the technology available today, many difficult obstacles such as the need to develop a user network and of a convenient communication medium had to be faced.

Gradually, the media used to organize the trips was changing from telephone to more flexible means such as the Internet, email and smartphones. Nowadays, many different carpooling platforms and services exist, but even today, they may be considered in their early stages because none has reached a critical mass of users.

Table 1 shows several features of different existing carpooling systems, including the most relevant security-related ones. In particular, we have chosen for this comparative analysis the representative systems: CarPooling [6], Blablacar [3], Amovens [1], ZimRide [28], [compartir.org](http://compartir.org) [9].

The main trust enforcing system in all these platforms is based on points given by users. However, bypassing this security system is quite easy because users who obtain a negative score, can create a new profile with new credentials and no points.

Apart from these practical platforms already in use, there are several papers that propose different solutions [12, 19, 23]. The work [5] shows an integrated system for the organization of carpooling service by using different technologies such as web, GIS and SMS. The authors of [21] propose a web platform to carpool.

**Table 1.** Comparison among several carpooling platforms

Platform	Social network	Privacy	Reputation system	Phone Cert.	Trust Alg.
Amovens [1]	yes	yes	yes	no	no
Blablacar [3]	yes	yes	yes	yes	no
CarPooling [6]	yes	no	yes	no	no
Compartir [9]	no	yes	no	no	no
ZimRide [28]	yes	yes	no	no	no
<b>Our system</b>	<b>yes</b>	<b>yes</b>	<b>yes</b>	<b>no</b>	<b>yes</b>

The paper [13] presents a carpooling architecture that uses a credit mechanism to encourage cooperation between users.

A more recent work is [8], where an algorithm to encourage carpooling is proposed based on assigning priority to users with positive feedbacks through a fuzzy logic scheme. Another paper, [14], defines a push service to promote carpooling through instant processing. Finally, another interesting proposal is [4], based on a secure multi-agent platform that focuses on the security services allowing both the mutual authentication between the users and the application components with the system.

The main aspect of our work is different from the aforementioned because it deals with the trust aspect of carpooling services through a combination of reputation measurement with privacy protection.

### 3 Carpooling Platform

The main objective of the proposed design is the increase of both usability and security. Thus, its key factors are user-friendliness and privacy.

The database schema that has been followed in this work uses the three entities mentioned below: Route, Vehicle and User. Table 2 shows all the relevant data about the Route entity that users create, and are stored in the database.

One of the main features is that users who publish their trips have their privacy fully protected. Unlike other carpooling platforms, in the described system, no user is allowed to access data such as email, phone or full name of others, unless he/she is authenticated on the platform and the algorithm for checking mutual trust returns a valid permission for him/her. In this case, the interested user can see all the data in detail. Otherwise, he/she can only send a request so that the receiver can decide whether the applicant is to be trusted or not.

The algorithm is based on trust relationships so that people who want to use the platform first need to authenticate in the platform through social networks such as Facebook, Twitter or Google+. In this way, the algorithm checks the existence of some chain of trust between the applicant and other users, based on the so-called rule of six degrees of separation [27].

Table 3 shows all the information necessary to analyze the system data and compute the degree of separation among different users in social networks, as stored in the database for each User Entity.

**Table 2.** Route entity details

Attribute	Description
idRoute	Unique Route identifier
idVehicle	Unique identifier of the Vehicle responsible for the Route
idUser	Unique identifier of the User responsible for the Route
From	Departure point of the Route
To	Destination point of the Route
Seats	Number of seats available in the vehicle
Price	Estimated cost of the Route
Properties	Route properties: animals allowed, smoking allowed, only women, etc
Date	Date of the Route
Repeat	If the Route is repeated, periodicity

**Table 3.** User entity details

Attribute	Description
idUser	Unique User identifier
idFacebook	Facebook User identifier
idTwitter	Twitter User identifier
idGooglePlus	Google Plus User identifier
idGCM	Google Cloud Messaging User identifier
Name	User real name
Phone	User phone number
Email	User email

Besides, the reputation gained through the use of the application is an influent factor, which is considered in the decision on whether carpooling with another person or not. In order to do this, at the end of every shared travel, the application asks both drivers and passengers to score the other users. Such scores are used in future trips so that seats offered by car drivers with good scores appear in better positions than others with lower scores. Also well-scored passengers have higher probability to have access to more details of drivers.

Finally, Table 4 shows the details regarding the Pairing Entity responsible for linking routes with users, as stored in the database.

Our overall system architecture used as an application model known as client-server. The client-server architecture of our system is showed in Fig. 1.

Its different elements are the following:

- **Client:** Mobile device used for the system.
- **Server:** Hosted in the cloud, and divided into two parts. On the one hand, the GCM server is the Google Cloud Messaging server that handles all the

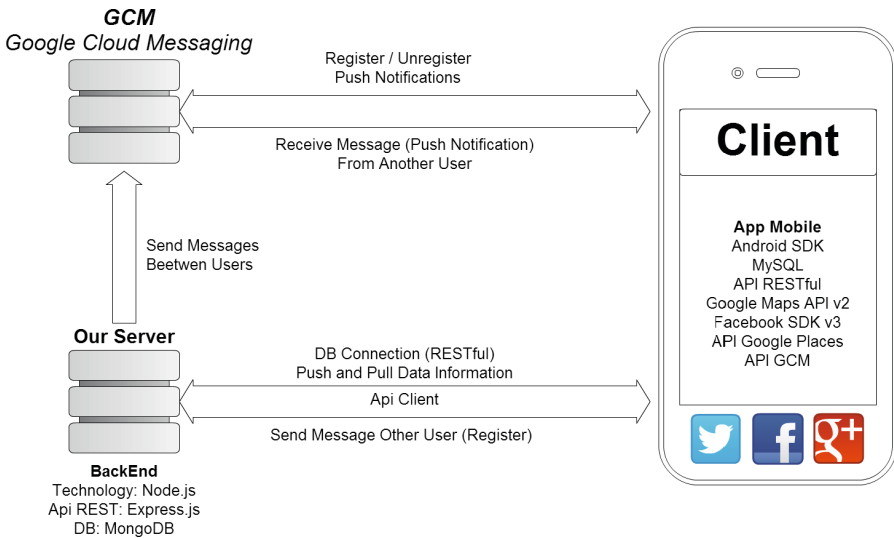
**Table 4.** Pairing entity details

Attribute	Description
idRoute	Unique Route identifier
idUser	Unique User identifier
State	Current state of the Route for a passenger who has applied: accepted, pending or declined
Type	Type of Route User: passenger or driver
Rating	Rating of the route: if passenger, the driver rating; otherwise, the list of passenger ratings

notifications and is responsible for sending the notification when the receiver clients are alive. On the other hand, the DB dedicated server is the server that stores in its DataBase all the data related to the users and system. It also serves as a gateway for sending notifications between the client and the GCM server.

The life cycle of the application is shown in Fig. 2, and explained below:

- **Step 1:** The user logs on the application through any of its social networks (Facebook, Twitter, Google+). Then, its data are saved in the DB server and the device is registered in the GCM server for notifications.
- **Step 2:** Once logged in, the user has two possibilities:
  1. The user can play the role of driver, and create new routes or manage existing routes. On the one hand, if a new route is created, all related



**Fig. 1.** Carpooling system architecture

information is stored in real time on the DB server. On the other hand, if the user is managing some old routes, first, the client informs the dedicated server to store the shares in the DB. Then, it also indicates to notify possible affected users about route changes in order to ask for acceptance/rejection to participate in the route, etc. The DB server sends the necessary information to the GCM server, so that this sends such notification to the involved clients when they are turned on and connected to the network.

2. The user can play the role of a passenger who tries to find some trip. Once a trip is found, the user sends a request to be a passenger on that route. This request must be answered positively or negatively by the creator of the route, through a notification sent within a period of time. Such a notification will go from the dedicated server to the GCM server because this is the responsible server for sending notifications to the clients.

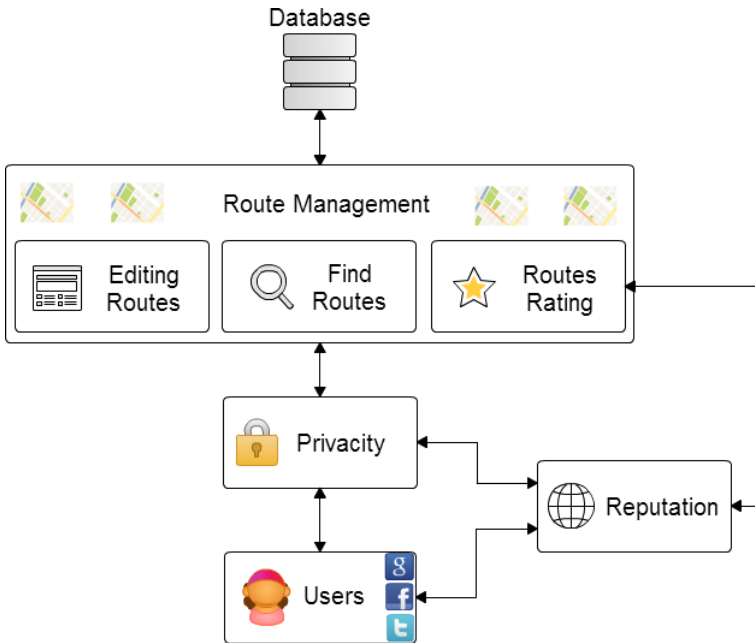


Fig. 2. Use model

The proposed scheme protects user privacy through limited and controlled access to user data, according to the trust level stated for the relationship between each pair of users. This trust level is got through the combination of direct scores and trust networks so that it provides the system with enough data to deduce whether people can trust each other or not. In this way, privacy is dealt with as one of the most important aspects of the proposed carpooling system.

A first approach to the development of a trust measurement algorithm that provides a value to each pair of users is based on the use of the PageRank algorithm to predict whether two people can rely each other. However, since this algorithm does not conform totally to the morphology of the specific problem, a second approach is also being used to complement it, based on Bayesian networks to know whether people can trust others. The refined trust measurement algorithm is available in the Android Application.

## 4 The Reputation Algorithm

No existing carpooling proposal offers the user a quantitative method, based on the theory of six degrees of separation, that can be used to decide whether another user is trustful to share a vehicle. Some of them even do not allow users to decide who may or may not apply your route. There are some proposed that the quantitative method based on the similarities among users [7, 16, 20], having to collect this information about the attributes and characteristics of each user. Our proposal aims to be simple for the user, without the user fills out some information about its attributes. With a simple click (that enables social login on the network that is previously registered) may be entered in our system and start using the platform. The main difference with our proposal is the proposed reputation algorithm. By using it, the user can use a quantitative measure to decide whether trusting another user. The algorithm is based on the theory of six degrees of separation and individual scores within our platform. Six degrees of separation is the theory that everyone is six or fewer steps away from each other in the world, so that a chain of 'a friend of a friend' statements can be made to connect any two people in a maximum of six steps. This number of steps may be reduced significantly by introducing the concept of social networks. Our application uses social networks when logging into the application to create network users to be used to interconnect with each other and provide a reliable measure of confidence. Through the use of social networks we can ensure that the six degrees of separation are reduced to only four. In particular, according to several researches on Facebook [2, 10, 26], the obtained average distance was 3.9, corresponding to intermediaries or 'degrees of separation', what shows that the world is even smaller than expected.

The reputation measurement is expressed as a number between 0 and 10, and is computed from the values given by each pair of users to inform about the reliability in each other. This measure is calculated by taking into account the two parameters mentioned above: degree of friendship and appreciation of other users on our platform. If the Facebook social network is used, the score given by the system with the degrees of friendship is broken down as follows:

- **7 points** if a user has only one degree of separation (i.e. it is direct friend).
- **6 points** if both users have a mutual friend.
- **4 points** if among those users there is a chain of friendship of more than two friends.
- **0 points** for the case that there is no chain friendliness.

Other users obtain the remaining scores from the assessments after sharing routes. When a route is completed, users who participated in it, can vote between **1 and 5** stars. Each passenger individually assesses the driver, and the driver individually assesses passengers. The weight is higher from driver to passengers than from passengers to driver, as the driver puts his/her vehicle available to the users. In order to account for the different ratings on a user, we use a simple arithmetic average. The metric taken into account for these ratings is as shown in Table 5.

**Table 5.** Impact of the ratings

Star rating	Driver rating impact	Passenger rating impact
1 star	-3 points	-2 points
2 stars	-1 point	0.25 points
3 stars	1.5 points	0.5 points
4 stars	2.5 points	2 points
5 stars	3 points	3 points

The maximum reputation score that a user can get is **10 points**. The system does not use a score below 0, so that 0 is the minimum score for any user. This valuation is dynamically calculated as a function of the friendship degree that a user has. It helps users to have a reliability measure about whether to trust another user of the platform. Besides, only users who have a valuation higher than **7.5 points** and/or users who have been accepted by the driver to make the route can see certain route data, such as the phone number or any other confidential data.

So the mathematical expression applied to the calculation of the reputation score in the algorithm is:

$$\left( lvFs + \frac{\sum_{i=1}^n rat[i]}{n} \right)$$

where:

- **lvFs**: Friendship level measured in points between 0 and 7, depending on the level of friendship each user has with other user, as explained above.
- **rat[]**: Each of the ratings a user has received both as driver and passenger, on the routes that has used. The points at which the rating of 1-5 stars are mapped.
- **n**: Total number of ratings that the user has received.

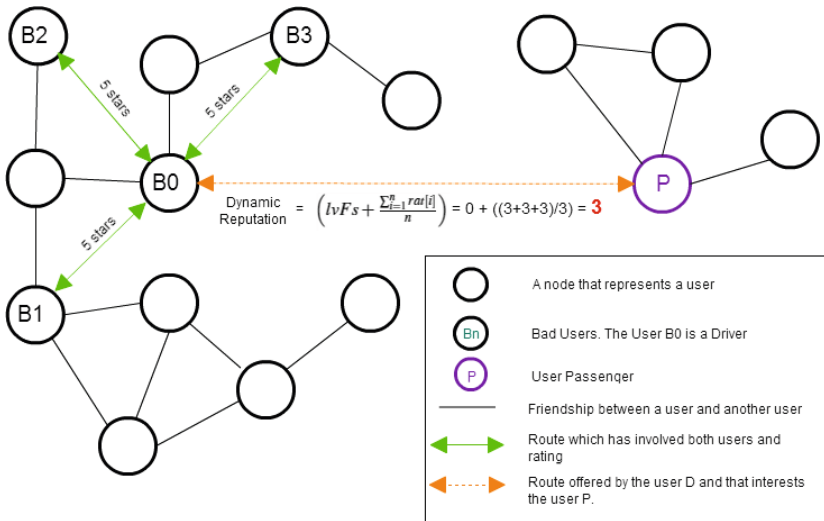


### 5 Security of the Scheme

Regarding the safety of the platform, Sybil attack [11] is one of notorious attacks in traditional carpooling systems. This type of attacks are hacking attacks on peer-to-peer networks where a malicious device illegitimately takes multiple identities by forging them. Due to the privacy-preserving environment of carpooling schemes, sybil vulnerability is generally hard to defend against.

In a Sybil attack the attacker subverts the reputation system of a peer-to-peer network by creating a large number of pseudonymous identities, using them to gain a disproportionately large influence. A reputation system’s vulnerability to a Sybil attack depends on how cheaply identities can be generated, the degree to which the reputation system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation system treats all entities identically.

A faulty node or an adversary may present multiple identities to a peer-to-peer network in order to appear and function as multiple distinct nodes. After becoming part of the peer-to-peer network, the adversary may then overhear communications or act maliciously. By masquerading and presenting multiple identities, the adversary can control the network substantially.



**Fig. 3.** Insufficient dynamic reputation rating between an bad user and a possible passenger.

In our case, for example, considering the following scenario:

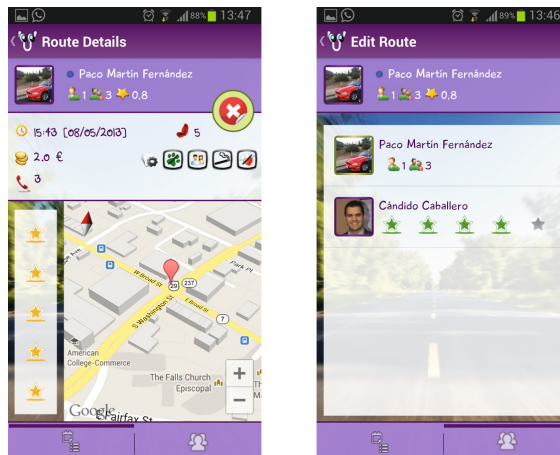
A ‘bad’ user ( $B0$ , See Fig. 3) sets up several bogus accounts in social media and the proposed system ( $B1, B2, B3$ , etc.). He/she then advertises a possible trip from  $X \mapsto Y \mapsto Z$ . For the first leg ( $X \mapsto Y$ ) he/she uses the bogus accounts

to claim that his/her vehicle is full and all these ‘passengers’ state that they get off at  $Y$ . He/she can then get excellent scores and increase his/her reputation. At some point, this will be so high that other normal users will be able trust him/her from leg  $Y \mapsto Z$ .

The proposed algorithm is protected against the attack described above because most of the score of our algorithm is preceded by confidence in degrees of friendship that binds each user to another user. Thus, if a user does not know (at all) another user, very high ratings of the latter in the system are not reliable enough for the former. It is remarkable that in our system the minimum reliability to trust another user is 7.5 points. At most, a user can have up to 3 points in relation to ratings for the routes that has done (see Fig. 3), far from the 7.5 points that are needed at least for that user to be reliable.

## 6 The Android Application

The proposed design has been embodied in an Android application that is already published in the Android Play Store under the name ‘Carpoolap’ (See Fig. 4).



**Fig. 4.** Passenger and driver rating screens

The Android application is developed for the versions 3.0 or higher of the operating system. APIs like Google Maps v3.0, Google Places, Google Cloud Messaging, etc., and Facebook SDKs 3.0 and libraries like Action Bar Sherlock used to use the functionality of the new versions of Android on older versions, were used. Autocomplete in address searches, Google Maps 3D Technology, design based on the latest versions of Android, push notifications with requests or responses of passengers or drivers, etc. are among the features of the Android Application.

Each user can see the routes he/she proposes as driver, and whether potential passengers exist for those routes. Besides, with colour codes, he/she can know the routes that each user has already made and the routes that have been confirmed by users. For the assessment of users participating in a route, after finishing it, each one can give a score. The different rating screens for the passenger and driver are shown in Fig. 4.

In order to deploy the carpooling platform, a server is also needed, so we developed one using javascript technologies by frameworks like ‘node.js’ and ‘express.js’. As a database for all the data centralized on this server, we decided to adopt a No SQL database, such as MongoDB [17]. We deployed our server on a micro instance of Amazon Web Services, specifically under Ubuntu machine with Amazon EC2 account.

## 7 Conclusions

By improving the confidence, security and reliability of carpooling systems it is possible to cause a greater use by the reticent society. In light of this, this paper proposes a solution to share free seats in vehicles that includes these enhancements. After the analysis of existing proposals and the review of related research works, it has not been detected any solution that uses the necessary degree of friendship to provide trust to the user. The presented proposal allows an assessment of users through their connections in social networks, based on the theory of six degrees of separations. It provides a reliable measure of confidence between users that enables the use of any carpool platform. A specific platform has been also designed and developed based on the proposed trusted algorithm. The whole code of the implementations both for the Android application and for the server are available as an Open Source Project in Github. For future work, the implementation of the solution for other mobile platforms is expected.

**Acknowledgments.** Research supported by the Spanish MINECO and the European FEDER Funds under projects TIN2011-25452, RTC-2014-1648-8, TEC2014-54110-R and IPT-2012-0585-370000, and the FPI scholarship BES-2012-051817.

## References

1. Amovens: <https://www.amovens.com>
2. Backstrom, L., Boldi, P., Rosa, M., Ugander, J., Vigna, S.: Four degrees of separation. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 33–42 (2011)
3. Blablacar: <https://www.blablacar.es>
4. Bonhomme, C., Arnould, G., Khadraoui, D.: Dynamic carpooling mobility services based on secure multi-agent platform. In: IEEE International Conference on Global Information Infrastructure and Networking Symposium, pp. 1–6 (2012)
5. Calvo, R.W., De Luigi, F., Haastrup, P., Maniezzoi, V.: A distributed geographic system for the daily car pooling problem. *Comput. Oper. Res.* **31**, 2263–2278 (2004)

6. Carpooling: [www.carpooling.es](http://www.carpooling.es)
7. Cho, S., Yasar, A., Knapen, L., Bellemans, T., Janssens, D., Wets, G.: A conceptual design of an agent-based interaction model for the carpooling application. In: The 1st International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications, pp. 801–807 (2011)
8. Collotta, M., Pau, G., Salerno, V.M., Scata, G.: A novel trust based algorithm for carpooling transportation systems. In: IEEE International Conference on Energy Conference and Exhibition, pp. 1077–1082 (2012)
9. Compartir.org: <http://compartir.org/>
10. Daraghmi, E.Y., Yuan, S.M.: We are so close, less than 4 degrees separating you and me! *Comput. Hum. Behav.* **30**, 273–285 (2014)
11. Douceur, J.R.: The Sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, p. 251. Springer, Heidelberg (2002)
12. Fellows, N.T., Pitfield, D.E.: An economic and operational evaluation of urban car-sharing. *Transp. Res. Part D: Transp. Environ.* **5**, 1–10 (2000)
13. Ferreira, J., Trigo, P., Filipe, P.: Collaborative car pooling system. In: International Conference on Sustainable Urban Transport and Environment (2009)
14. Fougères, A.J., Canalda, P., Ecarot, T., Samaali, A., Guglielmetti, L.: A push service for carpooling. In: IEEE International Conference on Green Computing and Communications, pp. 685–691 (2012)
15. Katzev, R.: Car sharing: a new approach to urban transportation problems. *Anal. Soc. Iss. Pub. Policy* **3**, 65–86 (2003)
16. Kuter, U., Golbeck, J.: SUNNY: a new algorithm for trust inference in social networks using probabilistic confidence models. In: Proceedings of the Conference on Artificial Intelligence AAI, pp. 1377–1382 (2007)
17. MongoDB: [www.mongodb.org](http://www.mongodb.org)
18. Oram, A.: *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology*. O’Reilly, Sebastopol (2001)
19. Prettenhaler, F.E., Steining, K.W.: From ownership to service use lifestyle: the potential of car sharing. *Ecol. Econ.* **28**, 443–453 (1999)
20. Pukhovskiy, N.V., Lepshokov, R.E.: Real-time carpooling system. In: International Conference on Collaboration Technologies and Systems, pp. 648–649 (2011)
21. Roberts, A., Pimentel, H., Karayevm, S.: CabFriendly: A Cloud-based Mobile Web App. Amazon’s EC2 (2011). <https://github.com/sergeyk/cabfriendly>
22. Steining, K., Vogl, C., Zettr, R.: The size of the market segment and revealed change in mobility behavior. *Transp. Policy* **3**, 177–185 (1996)
23. Talele, T., Pandi, G., Deshmukh, P.: Dynamic ridesharing using social media. *Int. J. AdHoc Netw. Syst.* **2**, 29 (2012)
24. Teal, R.: Carpooling: who, how and why. *Transp. Res. Part A* **21A**, 203–214 (1987)
25. Transport For London: [www.tfl.gov.uk/tfl/roadusers/congestioncharge](http://www.tfl.gov.uk/tfl/roadusers/congestioncharge)
26. Ugander, J., Karrer, B., Backstrom, L., Marlow, C.: The anatomy of the facebook social graph (2011). [arXiv:1111.4503](https://arxiv.org/abs/1111.4503)
27. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. *Nature* **393**, 440–442 (1998)
28. Zimride: <http://www.zimride.com/>